TABLE NO | | |

STUDENT ID NO | | | | | | | | | | |

MULTIMEDIA UNIVERSITY

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 3, 2018/2019

TSE3151 – SOFTWARE DESIGN

(All sections / Groups)

29 MAY 2019
9:00 am – 11:00 am
(2 Hours)

Question	Mark
A	
B	
C	
D	
Total	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature: _____

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 11 printed pages (including cover page) with 4 Sections only.

2. Attempt **ALL** questions in **SECTION A, SECTION B, SECTION C and SECTION D**. The distribution of the marks for each question is given.

3. Please write all your answers **CLEARLY** in the specific answer box provided for each question. Submit this question paper at the end of the examination.

Attempt ALL questions in SECTION A, B, C and D.

Section A (12.5 marks)

Consider the following Food example program design with "Decorator" Design Pattern.

```
interface Food
    { float getPrice() ;
      String getDescription() ;
    }
class Pizza implements Food
    { float getPrice()
        { return 50 ; }
      String getDescription()
        { return "Regular Pizza" ; }
    }
class Addition implements Food
    { Food f ;
      Addition(Food f)
        { this.f = f ; }
      float getPrice()
        { return f.getPrice() ; }
      String getDescription()
        { return f.getDescription() ; }
    }
class PlusOnion extends Addition
    { PlusOnion(Food f)
        { super(f) ; }
      float getPrice()
        { return super.getPrice() + 5 ; }
      String getDescription()
        { return super.getDescription() + ", with Onion"); }
    }
class PlusCorn extends Addition
    { PlusCorn(Food f)
        { super(f) ; }
      float getPrice()
        { return super.getPrice() + 7 ; }
      String getDescription()
        { return super.getDescription() + ", with Corn"); }
    }
// Etc.
    ... // In the program
    Food f = new PlusOnion(new PlusCorn(new Pizza())) ;
    ...
    System.out.println("You bought : "+f.getDescription()) ;
    System.out.println("Price in Shekels is : "+f.getPrice()) ;
```

Continued...

Among the name in the **Decorator** Design Pattern (DP) include *Component*, *ConcreteComponent*, *operation( )*, *Decorator*, *componentAggregationVariable*, ConcreteDecoratorA, and *ConcreteDecoratorB*.

Based on the above context, answer the following questions Q-A1 to Q-A2:

A1. Write a table to show the mapping of **Decorator** DP to the above Food example program.

(4 marks)

Continued...

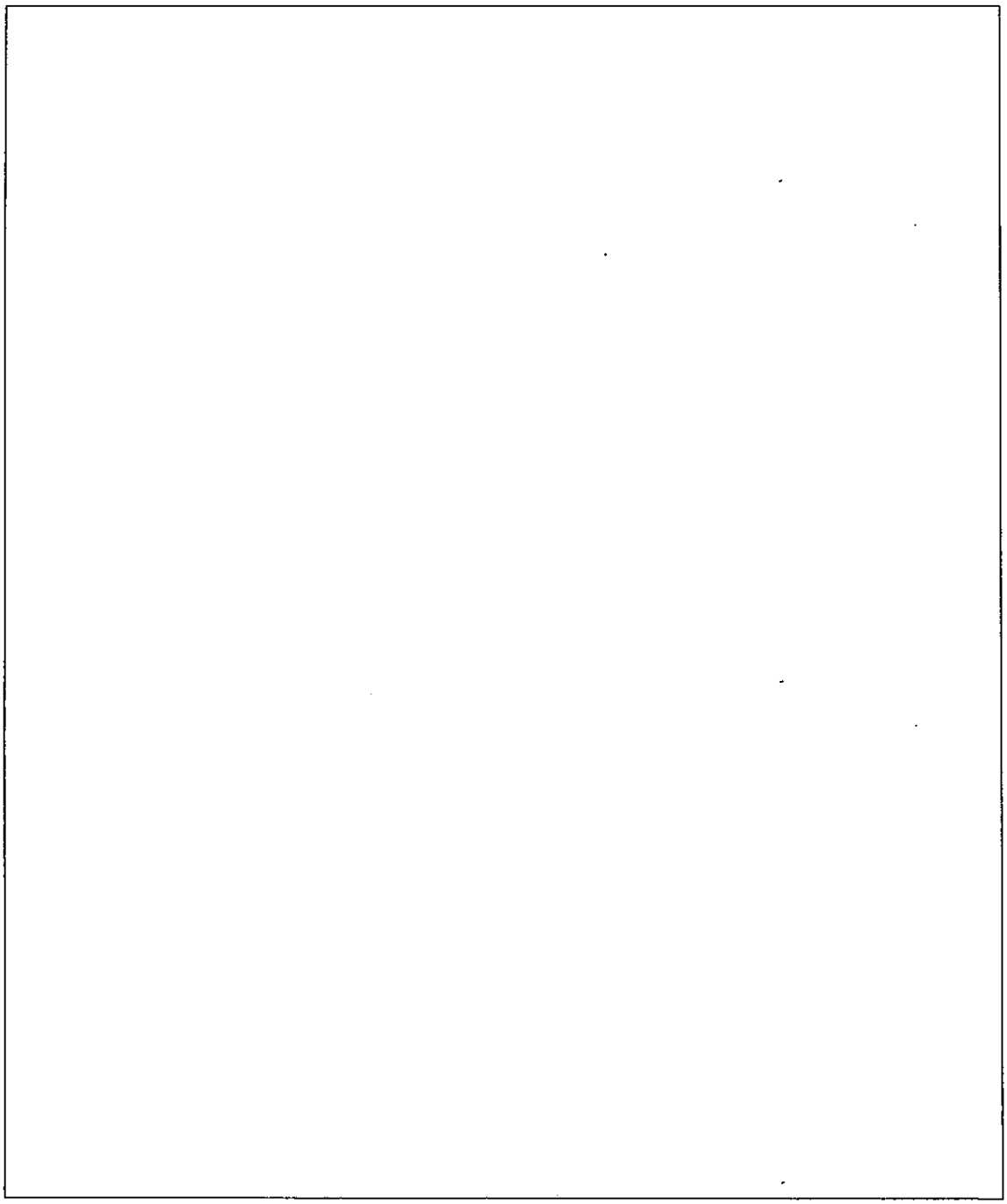
A2. Draw a class diagram for the typical Decorator design pattern. Among the name in the Decorator design pattern include *Component*, *ConcreteComponent*, *operation( )*, *Decorator*, *componentAggregationVariable*, ConcreteDecoratorA with *addedState* variable or attribute, and *ConcreteDecoratorB* with *AddedBehavior()* method.

(3.5 marks)

Continued...

A3. Draw a class diagram for the typical **Adapter** DP with the name in DP of *Adaptee*, *adapteeMethod( )*, *Target*, *targetMethod( )*, *Adapter*, *adapteeAggregationVariable*, and *Client*.

(5 marks)



Continued...

Section B (12.5 marks)

B1. You are asked to plan a journey by air from Kuala Lumpur in Malaysia to Shanghai in China.
Airline A is cheap, but involves flying to Bangkok, waiting there for five (5) hours and then flying on to Shanghai. Airline B is the most expensive, but offers a direct flight. Airline C has a package that is cheaper than that of airline A, but involves flying from Kuala Lumpur to Taipei and then to Shanghai via yet another airport in the Taiwan.

B1a. List FOUR factors besides price which will affect your choice of airline? These choices would subsequently be focused in the airline software design.

(2 marks)

B1b. Describe a decision scenario regarding your choice on the airline and route that meets the "wicked problem criteria". Explain your answer.

(4 marks)

Continued...

B2. Designing software is made more complex because we may be designing a sequence of actions. Design for at least 13 steps of instructions to help e-Shoppers access the e-Marketplace, create shopping carts for orders, and assign carts to an Assignee who will complete the purchase.

(6.5 marks)

Continued...

Section C (12.5 marks)

C1. Consider a physical apartment electrical grid design used with a software aided design system. Describe the following:
(C1a) THREE (3) viewpoints that might be needed in order to provide a full design description,
(C1b) FOUR (4) representations with examples that could be used for these design descriptions.

(3 + 2 marks)

C2. Suggest how you might represent the following viewpoints using in turn: text on its own; and a diagram on the program units (procedures) that make use of a particular data type in a program.

(2.5 marks)

Continued...

C3. Consider empirical research, which indicates that inexperienced programmers struggle with notations. Another RCT (randomized controlled trial) with human subjects, shows that, in C++, novices have particular trouble with lambda expressions. A large-scale survey of Java compiler errors from students across the world reinforces RCT findings on common typing and syntax mistakes, suggesting a coherence between field and lab studies.

There was also investigation found that blocks- and visualization-based programming can be slightly more beneficial than a purely text-based approach in certain regions of code, but not in other regions.

Based on the above scenario, answer questions Q-C3a to Q-C3b:

(C3a) Explain TWO reasons in favour of standardizing any particular form of design description.

(2 marks)

(C3b) Explain THREE reasons against standardizing the same form of description.

(3 marks)

Continued...

Section D (12.5 marks)

Consider a medium sized software system of CyberSecurity Education Software System (CyberSecEdu), which offer five different courses, each featuring a very hands-on approach. Introduction to Cybersecurity provides a broad introduction to the different components of cyber-enabled systems. Security issues and potential solutions within each component discussed set the tone for future courses.

Information Security and Assurance covers the topics of threat modeling, secure programming, applied cryptography, risk analysis, and the requirements of a compliance framework.

Information Security in System Administration provides prerequisite mathematical skills in such areas as propositional logic and graph theory, enabling students to draft mathematically provable security policies for perceived threats to cyber-enabled systems. The course also provides instruction in access control paradigms and operational tools that facilitate system administration.

Web Application Security focuses on advanced understanding of topics related to modern web-enabled distributed systems and their interaction, with a focus on the relevant security and privacy issues for such systems that enable consumer-centric services.

Computer Forensics covers disk forensics, network forensics, and malware analysis, with an emphasis on applying existing relevant tools to solve problems and the ability to create new tools to solve specific forensic or malware issues.

These courses are supported by curricula drawn from other relevant fields and adapted as prerequisites for the cybersecurity courses. The mathematical courses focus on mathematical cryptography and probabilistic and statistical analysis. The computer science courses cover various programming languages [e.g., C, C++, Python, XML, JavaScript, Structured Query Language (SQL), Bash, and others], data structures, algorithms, operating systems (e.g., Unix, Windows), databases, and web-enabled distributed systems.

Based on the above context, answer the following questions Q-D1 to Q-D3:

D1. Explain how **Abstract Factory** DP able to provide the solution to the above CyberSecEdu software system requirement.

(3 marks)

Continued...

D2. Draw a class diagram for the **typical Abstract Factory DP** with the name in DP of ***AbstractFactory***, ***createProductA()***, ***createProductB()***, ***ConcreteFactory1***, ***ConcreteFactory2***, ***AbstractProductA***, ***ProductA2***, ***ProductA1***, ***AbstractProductB***, ***ProductB2***, ***ProductB1***, and ***Client***. Note that this typical DP can be used to establish the mapping to the actual components of the software system when necessary.

(8.5 marks)

D3. Design patterns are a form of reuse. What are you reusing in design patterns for the abovementioned CyberSecEdu system?

(1 mark)

End of Paper